

BAB II

LANDASAN TEORI

A. Sistem Pakar

1. Pengertian Sistem Pakar

Kecerdasan Buatan (*Artificial Intelegant*) merupakan salah satu bagian dari ilmu komputer yang membuat agar komputer dapat melakukan pekerjaan seperti yang dilakukan manusia (Sri Kusumadewi, 2003). Kecerdasan buatan memiliki banyak bidang terapan diantaranya *Expert Sistem* (sistem pakar), *Natural Language Processing* (pemrosesan bahasa ilmiah), *Computer Visio* (mengintrepetasi gambar melalui komputer), *Intelligence Computer Aided Instruction* (tutor dalam melatih dan mengajar), *Speech Recognition* (pengenalan ucapan), *Robotics and Sensory Sistem* (robotika dan sistem sensor).

Sistem pakar adalah suatu sistem informasi yang berusaha mengadopsi pengetahuan dari manusia ke komputer, agar komputer dapat menyelesaikan masalah layaknya seorang pakar (Sri Kusumadewi, 2003). Sedangkan pengertian sistem informasi adalah kumpulan elemen yang saling berhubungan satu dengan yang lain untuk membentuk suatu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi tersebut (Budi Sutejo, 2006).

Pada dasarnya sistem pakar diterapkan untuk mendukung aktifitas pemecahan masalah. Beberapa aktifitas pemecahan masalah yang dimaksud seperti (Lestari, 2012):

a) Interpretasi

Membuat kesimpulan atau deskripsi dari sekumpulan data mentah.

Pengambilan keputusan dari hasil observasi, termasuk pengenalan ucapan, analisis citra, interpretasi sinyal, dll.

b) Prediksi

Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu. Contoh: prediksi demografi, prediksi ekonomi, dll.

c) Diagnosis

Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati diagnosis medis, elektronis, mekanis, dll.

d) Perancangan (desain)

Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu. Contoh: perancangan layout sirkuit, bangunan.

e) Perencanaan

Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu. Contoh: perencanaan keuangan, militer, dll.

f) Monitoring

Membandingkan hasil pengamatan dengan kondisi yang diharapkan. Contoh: *computer aided monitoring system*.

g) *Debugging*

Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.

Contoh: memberikan resep obat terhadap kegagalan.

h) Instruksi

Mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subyek.

Contoh: melakukan instruksi untuk diagnosis dan *debugging*.

i) Kontrol

Mengatur tingkah laku suatu environment yang kompleks. Contoh: melakukan kontrol terhadap interpretasi, prediksi, perbaikan dan monitoring kelakuan sistem.

Dengan sistem pakar, pengguna dapat memperoleh informasi yang berkualitas dengan mudah seperti halnya memperoleh dari para ahli di bidangnya. Selain itu, sistem pakar juga dapat membantu aktifitas para pakar sebagai asisten yang mempunyai pengetahuan yang dibutuhkan.

2. Kelebihan dan Kekurangan Sistem Pakar

Sistem pakar memiliki beberapa fitur yang merupakan kelebihannya (Rika Rosnelly, 2003), seperti:

- a. Meningkatkan ketersediaan (*increased availability*).
- b. Mengurangi biaya yang diperlukan untuk keahlian per satu orang pengguna.
- c. Sistem pakar menghasilkan solusi yang bersifat konsisten dibandingkan manusia yang terkadang berubah-ubah karena kondisi fisiknya seperti saat kelelahan.
- d. Sistem pakar menjelaskan detail proses penalaran yang dilakukan sehingga mendapatkan suatu kesimpulan.

- e. Sistem pakar relatif memberikan respon yang cepat dibandingkan seorang pakar.
- f. Sistem pakar dapat digunakan untuk mengolah data basis pengetahuan secara baik.
- g. Berperan sebagai pembimbing yang pintar, sistem pakar memberikan kesempatan pada pemakai untuk menjalankan contoh program dan menjelaskan proses penalaran yang benar.

Disamping memiliki kelebihan, sistem pakar juga mempunyai kekurangan.

Menurut M.Arhami (2005) kekurangan sistem pakar adalah sebagai berikut:

- a. Untuk mendapatkan pengetahuan tidaklah selalu mudah, karena kadang kala pakar dari masalah yang dibuat tidak ada, dan walaupun ada, kadang-kadang pendekatan yang dimiliki oleh pakar tersebut berbeda-beda.
- b. Untuk membuat suatu sistem pakar yang benar-benar berkualitas, cukup sulit dan membutuhkan biaya yang tidak sedikit untuk pengembangannya.
- c. Kadang kala sistem tidak menghasilkan sebuah keputusan.
- d. Sistem pakar perlu diuji ulang secara teliti sebelum digunakan, sehingga dalam hal ini faktor manusia tetaplah menjadi dominan.

3. Elemen Manusia Pada Sistem Pakar

Pengembangan sistem pakar dari awal hingga menghasilkan solusi akhir melibatkan peran serta 4 kelompok (Rika Rosnelly, 2003) diantaranya:

- a. Pakar (*expert*)

Pakar adalah individu yang memiliki pengetahuan khusus, pemahaman,

pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu. Selain itu seorang pakar, juga memiliki kemampuan untuk mengaplikasikan pengetahuannya dan memberikan saran serta pemecahan masalah pada domain tertentu.

b. Pembangun pengetahuan (*knowledge engineer*)

Pembangun pengetahuan adalah individu yang memiliki tugas menerjemahkan dan mempresentasikan pengetahuan yang diperoleh dari pakar, baik berupa pengalaman pakar dalam menyelesaikan masalah maupun sumber terdokumentasi lainnya ke dalam bentuk yang diterima oleh sistem. Dalam hal ini, pembangun pengetahuan mengintrepetasikan dan merepresentasikan pengetahuan dalam bentuk jawaban atas pertanyaan – pertanyaan yang diajukan pada pakar atau pemahaman, penggambaran analogis, sistematis, konseptual yang diperoleh dari membaca beberapa dokumen cetak seperti *text book*, jurnal, makalah, dan sebagainya.

c. Pembangun Sistem (*system engineer*)

Pembangun sistem adalah individu yang bertugas untuk merancang antar muka pemakai sistem pakar, merancang pengetahuan yang sudah diterjemahkan oleh pembangun pengetahuan ke dalam bentuk yang sesuai dan dapat diterima oleh sistem pakar dan mengimplementasikan ke dalam mesin inferensi. Selain itu, pembangun sistem juga bertanggung jawab apabila sistem pakar akan diintegrasikan dengan sistem komputerisasi lain.

d. Pemakai (*user*)

Banyak sistem berbasis komputer mempunyai susunan pemakai tunggal.

Hal ini berbeda dengan sistem pakar yang memungkinkan mempunyai beberapa kelas pemakai. Tabel 2.0 berikut menunjukkan beberapa contoh hubungan antara kelas pemakai, kepentingan pemakai dan fungsi sistem pakar.

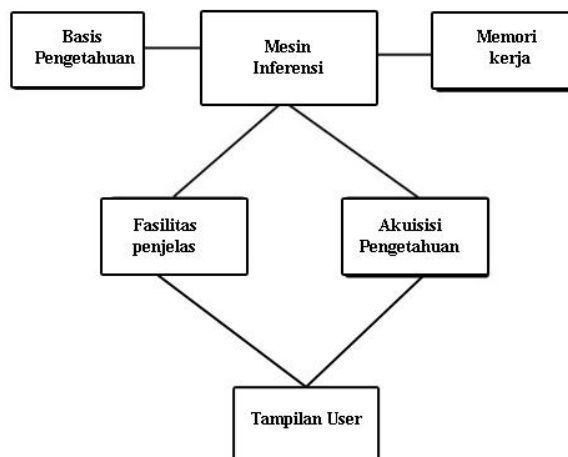
Tabel 2.0 Hubungan antar kelas pemakai, kepentingan pemakai dan fungsi sistem pakar

Pemakai	Kepentingan	Fungsi sistem pakar
Klien bukan pakar (masyarakat umum)	Mencari saran/nasehat, sarana belajar	Konsultan atau penasehat
Pembangun sistem dan pengetahuan	Memperbaiki/menambah basis pengetahuan, merancang sistem	Partner
Pakar	Membantu analisis rutin atau proses komputasi, mengklasifikasikan informasi, alat bantu diagnosa	Rekan kerja atau asisten

Sumber: Rika Rosnelly (2003)

4. Struktur Sistem Pakar

Menurut Rika Rosnelly (2003), adapun struktur sistem pakar dapat dilihat seperti pada gambar 2.1:



Gambar 2.1 : Struktur sistem pakar

Komponen yang terdapat dalam struktur sistem pakar ini adalah sebagai berikut:

a. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan berisi pengetahuan untuk pemahaman, formulasi dan penyelesaian masalah. Sistem pakar disusun atas dua elemen dasar yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui. Pada struktur sistem pakar diatas, *knowledge base* berfungsi untuk menyimpan pengetahuan dari pakar berupa rule / aturan (*if <kondisi> then <aksi>* atau dapat juga disebut *condition-action rules*).

b. Mesin Inferensi (*Inference Engine*)

Mesin Inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* atau *rule interpreter* (dalam sistem pakar berbasis kaidah). Komponen ini berisi mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah *processor* pada sistem pakar yang mencocokkan bagian kondisi dari *rule* yang tersimpan di dalam *knowledge base* dengan fakta yang tersimpan di *working memory*.

Ada beberapa teknik penalaran yang dapat digunakan salah satunya adalah *forward chaining*.

Foward chaining merupakan pencocokan fakta atau pernyataan dimulai dari bagian kiri (IF) atau dengan kata lain penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran (Sri Kusumadewi, 2003). Metode

ini sering disebut *data-driven* karena mesin inferensi menggunakan informasi yang ditentukan oleh pemakai untuk memindahkan ke seluruh jaringan dari logika 'AND' dan 'OR' sampai sebuah terminal ditentukan sebagai objek. Bila mesin inferensi tidak dapat menentukan objek maka akan meminta informasi lain. Aturan (*Rule*) dimana menentukan objek, membentuk lintasan (*path*) yang mengarah ke objek. Oleh karena itu, hanya satu cara untuk mencapai suatu objek adalah dengan memenuhi semua aturan.

Ada dua pendapat mengenai pelaksanaan metode ini. Pertama dengan cara membawa seluruh data yang didapat ke dalam sistem pakar. Kedua dengan membawa bagian penting-penting saja dari data yang didapat ke dalam sistem pakar. Cara pertama akan baik digunakan jika sistem pakar terhubung dengan proses otomatis dan dapat menerima seluruh data dari basis data. Namun cara kedua lebih efisien karena menghemat biaya dan waktu dengan mengambil data-data yang penting saja.

Contoh:

R1: IF A and C, THEN E

R2: IF D and C, THEN F

R3: IF B and E, THEN F

R4: IF B, THEN C

R5: IF F, THEN G

Fakta nya adalah: A benar dan B benar

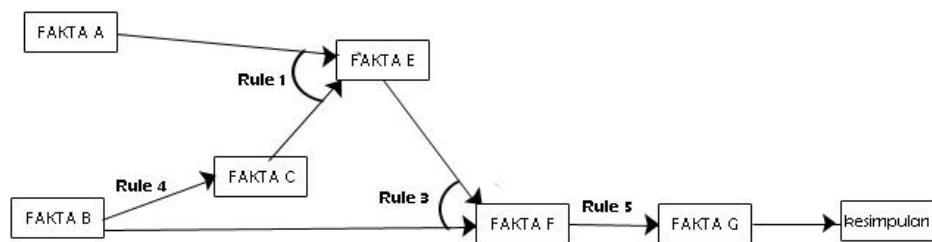
Langkah –langkahnya:

- 1) Dimulai dari R1 Karena C dan E tidak diketahui maka tidak diambil kesimpulan. Pencocokan lanjut di R2, ternyata di R2 juga tidak diambil

kesimpulan. Lakukan hal sama pada R3, kemudian pada R4 bernilai benar karena B diketahui benar.

- 2) Selanjutnya ke R5, namun tidak dapat diambil kesimpulan. Lalu kembali lagi ke atas, R1 bernilai benar karena A benar dan C benar maka F benar.
- 3) Lalu ke R2, karena D belum diketahui kebenarannya makanya tidak dapat diambil kesimpulan.
- 4) Lanjut pada R3, karena B dan F benar maka E juga benar.
- 5) Selanjutnya pada R5, karena F benar maka G benar. Sehingga dengan demikian G adalah kesimpulannya.

Diagram pohon *foward chaining* dapat dilihat pada Gambar 2.2:



Gambar 2.2 Diagram Pohon *Foward Chaining*

c. Memori Kerja (*Working Memory*)

Berguna untuk menyimpan fakta yang dihasilkan oleh mesin inferensi dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai *global database* dari fakta yang digunakan oleh aturan-aturan yang ada.

d. Fasilitas penjelasan (*Explanation facility*)

Menyediakan kebenaran dari solusi yang dihasilkan kepada pemakai.

e. Akuisisi pengetahuan (*Knowledge acquisition facility*)

Meliputi proses pengumpulan, pemindahan, dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi ke program komputer yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

f. Tampilan pemakai (*User Interface*)

Mekanisme untuk memberi kesempatan kepada pemakai dan sistem pakar untuk berkomunikasi antar muka yaitu dengan menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antar muka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai

5. Faktor Kepastian (*Certainty Factor*)

Dalam pembuatan sistem pakar diagnosa penyakit gangguan pernafasan, metode pengambilan kesimpulan yang digunakan adalah *Certainty Factor*. *Certainty Factor* merupakan bagian dari *Certainty Theory*, yang pertama kali diperkenalkan oleh Shorliffe Buchanan dalam pembuatan MYCIN (adalah aplikasi sistem pakar awal yang dirancang untuk mengidentifikasi bakteri yang menyebabkan infeksi berat) mencatat bahwa dokter sering kali menganalisa informasi yang ada dengan ungkapan seperti misalnya: mungkin, kemungkinan besar, hampir pasti. Untuk mengakomodasi hal ini tim MYCIN menggunakan *certainty factor (CF)* guna menggambarkan tingkat kepercayaan pakar terhadap masalah yang sedang dihadapi. CF atau faktor kepastian juga berguna untuk

mengatasi ketidakpastian dalam menentukan penyakit yang mempunyai gejala (*evidence*) yang sama. Dalam CF dikenalkan konsep *Measures of Belief (MB)* atau ukuran kepercayaan dan *Measures of Disbelief (MD)* atau ukuran ketidakpercayaan..

Dalam memberikan ukuran MB, MD, dan CF, tim MYCIN mempunyai parameter untuk menunjukkan ukuran kepercayaan. Berikut tabel 2.1 aturan nilai-nilai kepercayaan dan tabel 2.2 nilai interpretasi untuk MB dan MD yang diberikan oleh MYCIN.

Tabel 2.1 Aturan nilai-nilai kepercayaan

Kepercayaan	CF
Tidak Pasti	-1,0 sampai -0,79
Hampir Tidak Pasti	-0,8 sampai -0,59
Kemungkinan Tidak	-0,6 sampai -0,39
Mungkin Tidak	-0,4 sampai -0,19
Tidak Tahu	-0,2 sampai 0,2
Mungkin	0,4 sampai 0,59
Kemungkinan Besar	0,6 sampai 0,79
Hampir Pasti	0,8 sampai 0,89
Pasti	0,9 sampai 1,0

Tabel 2.2 Nilai interpretasi untuk MB dan MD

Kepercayaan	MB / MD
Tidak tahu	0 – 0,29
Mungkin	0,3 – 0,49
Kemungkinan Besar	0,5 – 0,69
Hampir Pasti	0,7 – 0,89
Pasti	0,9 – 1,0

Certainty Factor menggunakan ukuran kepercayaan (MB) dan ukuran ketidakpercayaan (MD) dalam mengkombinasikan beberapa *evidence* untuk menentukan nilai CF suatu hipotesis. Konsep ini kemudian di formulasikan ke dalam rumus 2.0 (Sri Kusumadewi, 2003):

Rumus 2.0:

$$CF [h,e] = MB [h,e] - MD [h,e]$$

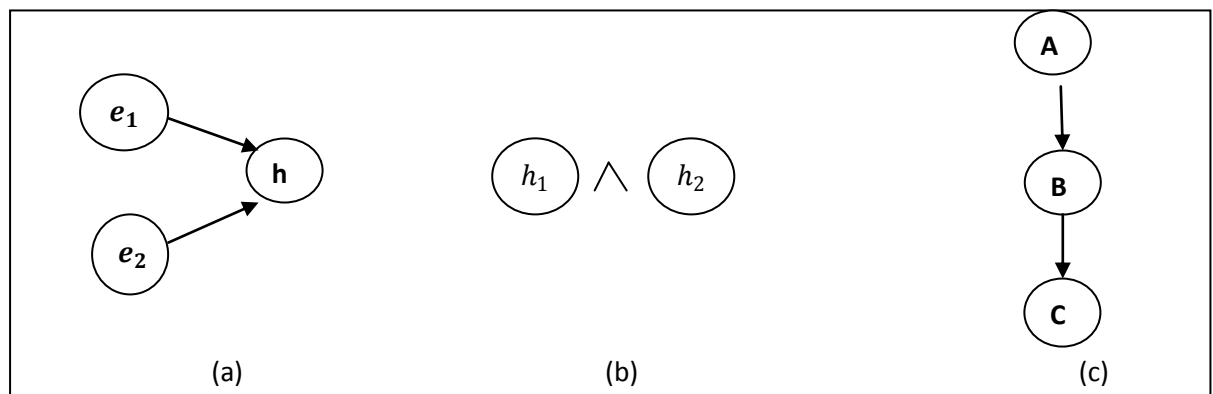
Keterangan:

$CF [h,e]$ = faktor kepastian dari hipotesis h yang dipengaruhi oleh *evidence* e . Besarnya CF berkisar antara -1 sampai dengan 1
Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1 menunjukkan kepercayaan mutlak.

$MB [h,e]$ = ukuran kepercayaan terhadap hipotesis h , jika diberikan *evidence* e (antara 0 dan 1)

$MD [h,e]$ = ukuran ketidakpercayaan terhadap hipotesis h , jika diberikan *evidence* e (antara 0 dan 1)

Selanjutnya, pada *certainty factor* ada beberapa kemungkinan kombinasi aturan ketidakpastian yang dapat terjadi antara lain dapat dilihat pada Gambar 2.3



(Sri Kusumadewi, 2003)

Gambar 2.3 Kombinasi Aturan Ketidakpastian

- 1) Beberapa *evidence* dikombinasikan untuk menentukan CF dari suatu hipotesis seperti terlihat pada gambar 2.3(a). Jika e_1 dan e_2 adalah gejala, maka

Rumus 2.1:

$$\begin{aligned} MB[h, e_1 \wedge e_2] &= \begin{cases} 0 & , MD[h, e_1 \wedge e_2] = 1 \\ MB[h, e_1] + MB[h, e_2] \cdot (1 - MB[h, e_1]) & , \text{lainnya} \end{cases} \\ MD[h, e_1 \wedge e_2] &= \begin{cases} 0 & , MB[h, e_1 \wedge e_2] = 1 \\ MD[h, e_1] + MD[h, e_2] \cdot (1 - MD[h, e_1]) & , \text{lainnya} \end{cases} \end{aligned}$$

- 2) CF dihitung dari kombinasi beberapa hipotesa seperti yang terlihat pada gambar

2.3 (b)

Jika h_1 dan h_2 adalah hipotesis maka

Rumus 2.2:

$$\begin{aligned} MB[h_1 \wedge h_2, e] &= \min(MB[h_1, e], MB[h_2, e]) \\ MB[h_1 \vee h_2, e] &= \max(MB[h_1, e], MB[h_2, e]) \\ MD[h_1 \wedge h_2, e] &= \min(MD[h_1, e], MD[h_2, e]) \\ MD[h_1 \vee h_2, e] &= \max(MD[h_1, e], MD[h_2, e]) \end{aligned}$$

- 3) Beberapa aturan saling bergandengan, ketidakpastian dari suatu aturan menjadi input untuk aturan yang lainnya seperti yang terlihat pada gambar 2.3 (c) maka

Rumus 2.3:

$$MB[h, s] = MB'[h, s] * \max(0, CF[s, e])$$

Dengan $MB'[h, s]$ adalah ukuran kepercayaan h berdasarkan kepercayaan penuh terhadap validitas s . (Sri Kusumadewi, 2003).

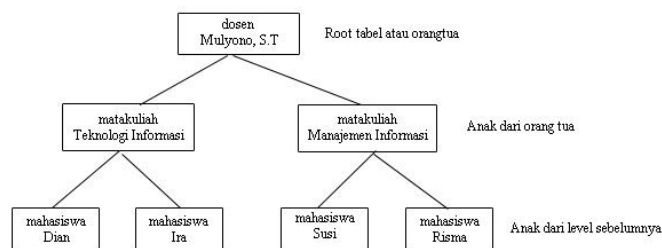
B. Basis Data Relasional (*Relation Database*)

Basis data merupakan komponen terpenting dalam membangun sebuah sistem informasi karena sebagai tempat untuk menampung dan mengorganisasi seluruh data yang ada dalam sistem. Menurut Connolly dan Bagui (2003), basis data

adalah suatu penyimpanan data yang tersusun atas sekumpulan data-data yang secara logika saling terkait dan dirancang untuk memenuhi kebutuhan informasi perusahaan. Menurut Ahmad dan Wahyu (2008) ada beberapa model basis data:

1. Model Basis Data Hirarki

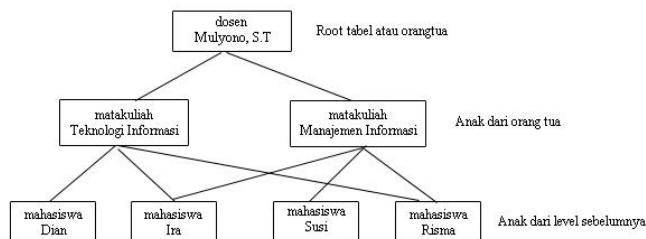
Pada model ini menggunakan pola hubungan orang tua - anak. Lapisan paling atas bertindak sebagai orang tua dari segmen yang tepat berada di bawahnya. Segmen yang berada di bawah dari suatu segmen lainnya merupakan anak dari segmen yang ada di atasnya. Setiap orang tua bisa memiliki hubungan dengan satu atau beberapa anak namun setiap anak hanya bisa memiliki satu orang tua. Pada gambar 2.4 adalah contoh model basis data hirarki



Gambar 2.4 Model Basis Data hirarki

2. Basis Data jaringan

Model ini menyerupai model hirarki, perbedaanya terletak pada suatu segmen anak pada model jaringan bisa memiliki lebih dari satu orang tua. Gambar 2.5 adalah model basis data jaringan.

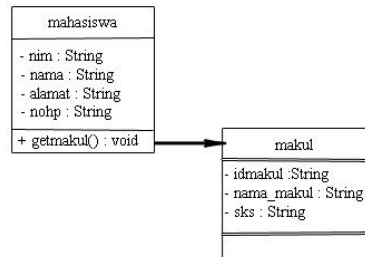


Gambar 2.5 Model Basis Data Jaringan

3. Basis Data Berorientasi Objek

Suatu model basis data yang mana data didefinisikan, disimpan dan diakses menggunakan pemrograman berorientasi objek.

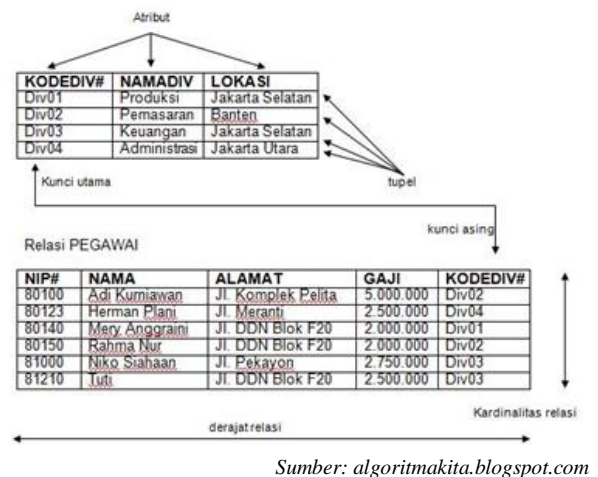
Pada gambar 2.6 adalah model basis data berorientasi objek



Gambar 2.6 Model Basis Data Berorientasi Objek

4. Basis Data Relasional

Model basis data relasional adalah sistem yang paling banyak digunakan karena struktur logikanya yang sederhana. Pada model relasional, seluruh data disusun secara logikal dalam relasi-relasi atau tabel. Setiap relasi terdiri dari baris dan kolom, dan kolom dari relasi yang diberi nama tertentu disebut atribut. Sedangkan baris dari relasi disebut tuple dan setiap tuple memiliki satu nilai untuk setiap atribut. Basis data yang tabel-tabelnya saling berhubungan dikatakan memiliki relasi. Karena tidak ada relasi yang memiliki dua tuple yang sama, maka setiap baris dapat diidentifikasi secara unik dengan menggunakan kunci utama (*primary key*). Munculnya sebuah atribut dalam beberapa relasi dapat merepresentasikan hubungan antar tuple dari relasi-relasi tersebut. Model Basis Data Relasional dapat dilihat pada gambar 2.7



Gambar 2.7 Model Basis Data Relasional

Basis data adalah kumpulan datanya, sedangkan program pengelolanya berdiri sendiri dalam satu paket program komersial untuk membaca data, mengisi data, menghapus data dan melaporkan data dalam basis data. Salah satu contoh software DBMS (*Database Management System*) adalah MySQL. Sedangkan untuk pemodelan data dalam basis data menggunakan model *Entity Relationship Diagram* atau disebut ERD.

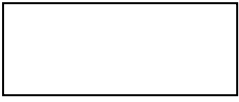
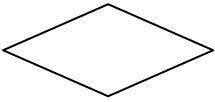
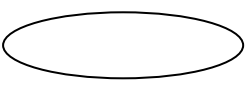

a) *Entity Relationship Diagram* (ERD)

Pengertian ERD Menurut Jogiyanto (2001) adalah suatu komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang mempresentasikan seluruh fakta dari dunia nyata yang ditinjau. ERD dibagi menjadi 2 yakni *Entity Relationship Diagram Logical Data Model* dan *Entity Relationship Diagram Physical Data Model*. ERD *Logical Data Model* adalah konsep ERD yang mana data merepresentasikan sebuah kenyataan, dimasukkan ke dalam sebuah pemrosesan logika dan dapat menghasilkan informasi. Sedangkan ERD

Physical Data Model adalah konsep ERD yang menjelaskan bagaimana data di simpan pada media penyimpanan (*storage*) dalam suatu susunan.

ERD memiliki beberapa komponen-komponen diantaranya dapat dilihat pada tabel 2.3

Tabel 2.3 Komponen – komponen ERD

Simbol	Nama	Keterangan
	Entitas	Entitas adalah sebuah objek yang ada dan dapat dibedakan dengan objek lain.
	Relasi	Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda
	Atribut	Atribut berfungsi mendiskripsikan karakter entitas
	Garis	Garis berfungsi sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut

Dalam ERD terdapat 3 jenis relasi, antara lain:

1) Satu ke satu (1-1)

Setiap *record* di entitas pertama hanya dapat berelasi dengan satu *record* di entitas kedua, begitu pula sebaliknya.

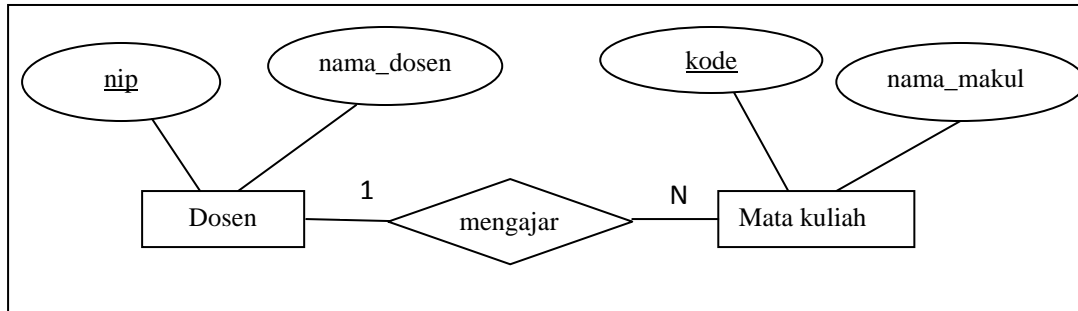
2) Satu ke banyak (1-N)

Setiap *record* di entitas pertama dapat berelasi dengan banyak *record* pada entitas kedua. Namun satu record pada entitas kedua hanya dapat berelasi dengan satu *record* saja pada entitas pertama.

3) Banyak ke banyak (N-N)

Banyak *record* di entitas pertama dan kedua dapat saling berelasi satu sama lain.

Contoh salah satu relasi ERD bisa dilihat pada gambar 2.8



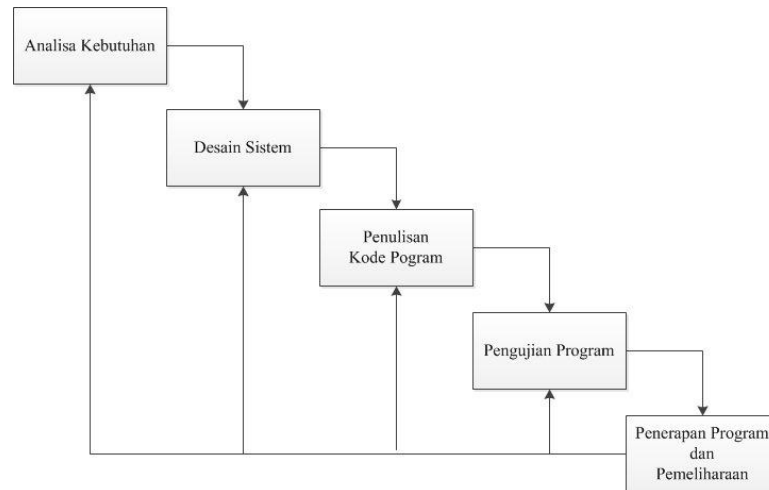
Gambar 2.8 Contoh relasi ERD

C. Jenis – Jenis Metode Pengembangan Sistem

Dalam membangun suatu sistem informasi diperlukan sebuah cara atau metode yang dijadikan sebagai panduan untuk mendapatkan sistem yang diharapkan. Beberapa metode pengembangan sistem adalah *Prototype*, *Incremental*, *Code-and-fix*, *Star Lifecycle*, *Spiral* dan *waterfall*.

Metode *Waterfall* atau dapat disebut sebagai model air terjun adalah satu metode dalam pengembangan sistem yang mempunyai ciri khas pengerjaan setiap fase dalam *waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya.

Menurut Kadir (2003), model waterfall mempunyai langkah –langkah sebagai berikut seperti terlihat pada Gambar 2.9:



Gambar 2.9 Model Waterfall

Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam model tersebut:

a. Analisa Kebutuhan

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dilakukan melalui wawancara dan studi literatur. Sistem analis akan menggali informasi sebanyak-banyaknya dari pemakai sehingga akan tercipta sebuah sistem komputer yang melakukan tugas-tugas yang diinginkan oleh pemakai tersebut. Tahapan ini akan menghasilkan dokumen pemakai *recruitment* atau dikatakan sebagai data yang berhubungan dengan keinginan pemakai dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menerjemahkan ke dalam bahasa pemrograman

b. Desain Sistem.

Tahapan dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (*data flow diagram*), serta *use case diagram*.


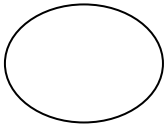

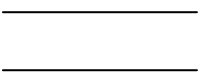
1) *Data Flow Diagram (DFD)*

Data Flow Diagram (DFD) adalah diagram yang menggunakan notasi simbol untuk menggambarkan arus data sistem (Jogiyanto Hartono, 2005).

DFD merupakan gambaran sistem secara logika dari input sampai menghasilkan output dan tidak terikat oleh perangkat keras, perangkat lunak, dan organisasi file. DFD berfungsi memudahkan pemakai untuk memahami bagaimana proses kerja sistem yang akan dibuat.

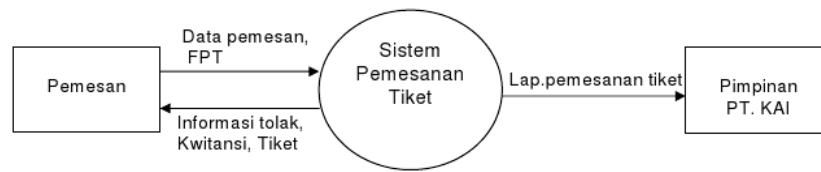
Simbol – simbol yang digunakan dalam DFD terlihat pada tabel 2.4

Tabel 2.4 Simbol – simbol DFD

Simbol	Nama	Keterangan
	Entitas Eksternal	Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
	Proses	Kegiatan dari hasil suatu arus data yang masuk dalam proses dihasilkan arus data yang akan keluar atau mengubah input menjadi output.
	Aliran data (<i>data flow</i>)	Data mengalir melalui sistem, dimulai dengan sebagian input dan diubah menjadi output.
	Penyimpanan data (<i>data storage</i>)	Data disimpan untuk keperluan berikutnya

Adapun tingkatan-tingkatan dari suatu DFD adalah sebagai berikut (Kendall, 2006) adalah Diagram level 0, diagram level 1, diagram level n.

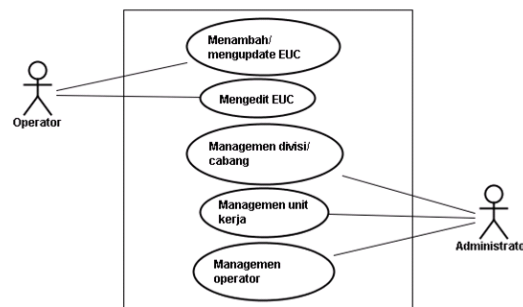
Gambar 2.10 adalah salah satu contoh DFD



Gambar 2.10 Contoh DFD

2) Use Case Diagram

Use Case Diagram adalah model fungsional sebuah sistem yang menggunakan aktor dan *use case*. Aktor berfungsi menjelaskan seseorang atau sesuatu yang berinteraksi dengan sistem. Menurut Hendri et al (2008) *use case* adalah fasilitas atau fungsi-fungsi yang disediakan untuk pemakainya. Pada gambar 2.11 berikut adalah contoh *use case*



Gambar 2.11 Contoh *Use Case*

c. Penulisan Kode Program

Penulisan kode program atau *coding* merupakan penerjemahan *design* dalam bahasa yang dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh pemakai. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah

pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian diperbaiki.

d. Pengujian Program

Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

e. Penerapan Program dan Pemeliharaan

Sistem yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut karena mengalami kesalahan karena sistem harus menyesuaikan dengan lingkungan baru (periperal atau sistem operasi baru), atau karena pelanggan membutuhkan perkembangan fungsional sehingga diperlukan pemeliharaan.

D. Analisis Penyakit Gangguan Pernafasan

Dalam mendiagnosa suatu penyakit, maka yang harus dilakukan pertama kali adalah mencari tahu gejala-gejala yang sering muncul dari penyakit tersebut. Meskipun hanya berupa gejala klinis (gejala yang dapat di lihat maupun yang dirasakan penderita), namun dokter masih dapat mengambil suatu kesimpulan berupa penyakit yang diderita pasien. Tetapi ada kalanya perlu pemeriksaan lanjutan untuk jenis penyakit tertentu.

Berikut ini adalah beberapa jenis penyakit gangguan sistem pernafasan beserta gejalanya (Iman Soemantri, 2007):

1. Faringitis

Faringitis atau biasa disebut radang tenggorokan adalah radang pada faring karena infeksi sehingga timbul rasa nyeri pada waktu menelan makanan ataupun kerongkongan terasa kering. Gangguan ini disebabkan oleh infeksi bakteri atau virus. Bakteri yang biasa menyerang penyakit ini adalah *Streptococcus pharyngitis*.

Gejala:

- Batuk berdahak
- Flu
- Suara serak atau sengau
- Tenggorokan terasa nyeri ketika menelan makanan
- Kerongkongan terasa kering
- Gatal pada tenggorakan

Pada kasus ringan karena infeksi virus tidak harus ke dokter, cukup dengan pemberian obat penghilang rasa sakit atau demam. Namun pada kasus tertentu karena virus, perlu dilakukan pemeriksaan lanjutan oleh dokter yaitu spatel lidah untuk mengetahui keadaan tonsil dan hiperemis, Pemeriksaan biopsi.

2. Asma

Asma adalah kelainan penyumbatan saluran pernapasan yang disebabkan oleh alergi seperti debu, bulu, ataupun rambut. Asma merupakan penyempitan saluran pernapasan utama pada paru-paru. Kelainan ini tidak menular dan bersifat genetis

atau bawaan seseorang sejak lahir. Kelainan ini juga dapat kambuh jika suhu lingkungan cukup rendah atau keadaan dingin, udara kotor, alergi, dan stres (tekanan psikologis). Hampir separuh jumlah penderita mendapat asma karena alergi ataupun sistem pernafasan yang terlalu sensitif terhadap debu, obat, makanan, dan minuman. Pola hidup tidak sehat turut mempengaruhi timbulnya penyakit asma, seperti merokok dan stress.

Gejala:

- Kesulitan bernafas karena sesak nafas
- Nyeri pada bagian dada
- Lebih sensitive terhadap alergi seperti udara dingin, debu, atau jenis makanan tertentu.
- Batuk-batuk hanya pada malam hari dan cuaca dingin.
- Badan terlihat letih dan lesu serta kurang bersemangat.
- Pilek
- Nafas berbunyi (mengi)

Asma merupakan penyakit kronis yang sulit disembuhkan. Meski begitu dengan tindakan pengobatan ataupun terapi yang dilakukan dengan teratur dapat meningkatkan kembali fungsi paru-paru secara maksimal sehingga meminimalisir serangan asma. Obat-obatan bagi penderita asma ada dua macam yakni obat sebagai controller atau pengendali asma biasanya diberikan setiap hari sebagai pencegah bila serangan asma sudah cukup berat. Obat reliever atau pereda serangan asma, obat ini hanya diberikan saat terjadi serangan asma. Untuk mencegah terjadinya serangan asma dengan menghindarkan alergen atau faktor

pencetus yang membuat alergi, gantilah sprei dan gorden seminggu sekali, menghindari penggunaan karpet karena menjadi tempat menempelnya debu.

3. Emfisema

Emfisema adalah penyakit pada paru-paru yang ditandai dengan pembengkakan pada paru-paru karena pembuluh darahnya kemasukan udara. Emfisema disebabkan hilangnya elastisitas alveolus. Emfisema membuat penderita sulit bernafas.

Gejala:

- Batuk kronis
- Tubuh mudah lelah ketika melakukan aktifitas fisik
- Susah tidur
- Kesulitan bernafas karena sesak nafas
- Nafsu makan berkurang
- Perubahan fisik pada bagian dada yang terlihat lebih cembung ke depan
- Nyeri pada bagian dada

Tindak lanjut yang dapat dilakukan jika terkena penyakit ini adalah dengan minum obat-obatan bronkodilator dan mukolitik agar dahak mudah dikeluarkan, juga dengan melakukan terapi oksigen. Pemeriksaan lanjutan: rontgen dada, pemeriksaan spirometri dan gas arteri.

4. Bronkitis

Bronkitis berupa peradangan pada selaput lendir dari saluran bronkial. Peradangan-peradangan tersebut dapat terjadi karena berbagai hal, di antaranya karena infeksi oleh mikroorganisme. Tetapi pada penderita yang mempunyai

penyakit menahun (seperti jantung ataupun paru-paru) dan pada usia lanjut bronkitis menjadi serius.

Gejala:

- Batuk berdahak.
- Kesulitan bernafas karena sesak nafas
- Susah tidur
- Timbul warna kemerahan pada wajah, telapak tangan, dan selaput lendir.
- Mudah lelah ketika melakukan aktifitas fisik
- Dahak keluar dalam jumlah besar
- Dahak berwarna kuning atau kemerahan

Tindakan yang dapat dilakukan untuk mencegah serangan yaitu istirahat yang cukup, menghindari polusi lingkungan, minum air putih sampai 3000ml/hari atau 8 gelas, latihan nafas dalam. Untuk pemeriksaan lanjutan dengan melakukan tes sinar X dada, tes fungsi paru, sputum, dan tes bronkogram.

5. Tuberculosis (TBC)

TBC adalah penyakit yang disebabkan oleh bakteri *Mycobacterium tuberculosis*. Bakteri ini dapat menyerang seluruh organ tubuh manusia, namun yang paling sering diserang adalah paru-paru (maka secara umum sering disebut sebagai penyakit paru-paru/TB Paru-paru). Bakteri ini menyerang paru-paru sehingga pada bagian dalam alveolus terdapat bintil-bintil. Penyakit ini menyebabkan proses difusi oksigen yang terganggu karena adanya bintik-bintik kecil pada dinding alveolus.

Gejala:

- Batuk lebih dari 2 minggu kadang disertai darah
- Kesulitan bernafas karena sesak nafas
- Nafsu makan yang menurun dan berat badan yang menurun
- Demam yang cukup lama
- Nyeri pada dada
- Sakit kepala
- Susah tidur
- Urine berubah warna menjadi keruh atau kemerahan

Tindakan yang dapat dilakukan adalah dengan memeriksakan diri ke dokter agar dilakukan tes dahak penderita lewat mikroskopis, rontgen, dan pemeriksaan kultur sputum.

6. Pneumonia

Pneumonia atau *Logensteking* adalah penyakit radang paru-paru yang disebabkan oleh virus *Diplococcus pneumoniae*. Pneumonia dapat juga disebabkan oleh iritasi kimia atau fisik dari paru-paru atau sebagai akibat dari penyakit lainnya, seperti kanker paru-paru atau terlalu banyak minum alkohol.

Gejala

- Demam tinggi
- Batuk berdahak
- Nyeri pada bagian dada
- Kesulitan bernafas karena sesak nafas
- Menurunnya tingkat kesadaran
- Bibir dan kuku membiru
- Dahak berwarna kehijauan dan kental seperti lendir

Tindakan pencegahan yang dapat dilakukan adalah menghindari faktor pencetus yang membuat penyakit semakin parah seperti hindari asap rokok, tidak minum minuman beralkohol, melakukan pola hidup sehat.

E. Jaminan Kualitas Sistem

Jaminan kualitas sistem adalah aktifitas pelindung yang diaplikasikan pada seluruh proses sistem. Tujuannya adalah untuk memberikan data yang diperlukan oleh manajemen untuk menginformasikan masalah kualitas sistem, sehingga dapat memberikan kepastian bahwa kualitas sistem dapat memenuhi sasaran.

Dalam menentukan kualitas suatu sistem, dibutuhkan suatu aspek ukuran yang bisa menjadi acuan seberapa puaskah pemakai terhadap penggunaan sistem yang dibuat. Komponen yang digunakan untuk mengukur seberapa baik suatu sistem tersebut sehingga pemakai merasa puas adalah sebagai berikut *International Standart Organization (ISO) 9241:11, 1998*):

- a. Kemudahan (*learnability*) didefinisikan seberapa cepat pemakai mahir dalam menggunakan sistem serta kemudahan dalam penggunaan menjalankan suatu fungsi serta apa yang pemakai inginkan dapat mereka dapatkan.
- b. Efisiensi (*efficiency*) didefinisikan sebagai sumber daya yang dikeluarkan guna mencapai ketepatan dan kelengkapan tujuan.
- c. Mudah diingat (*memorability*) didefinisikan bagaimana kemampuan pemakai mempertahankan pengetahuannya setelah jangka waktu tertentu,

kemampuan mengingat didapatkan dari peletakkan menu yang selalu tetap.

- d. Kesalahan dan keamanan (*errors*) didefinisikan berapa banyak kesalahan-kesalahan apa saja yang dibuat pemakai, kesalahan yang dibuat pemakai mencakup ketidaksesuaian apa yang pemakai pikirkan dengan apa yang sebenarnya disajikan oleh sistem.
- e. Kepuasan (*satisfaction*) didefinisikan kebebasan dari ketidaknyamanan, dan sikap positif terhadap penggunaan produk atau ukuran subjektif sebagaimana pemakai merasa tentang penggunaan sistem.

F. Pengukuran (Penskoran)

Suatu sistem yang akan diluncurkan perlu dilakukan pengukuran kualitas sistem salah satunya bisa menggunakan kuesioner yang dibagikan kepada pemakai. Menurut Syaiduffin azwar (2007) pengukuran adalah suatu prosedur pemberian angka terhadap atribut atau variabel suatu kontinum.

Hasil kuesioner dapat dihitung dengan menggunakan rumus 2.4, namun sebelum dilakukan perhitungan terlebih dahulu kriteria penilaian dalam kuesioner diberikan bobot. Tabel 2.5 adalah bobot skor yang diberikan syaifuddin azwar (2007)

Tabel 3.22 Penskoran Kuisisioner

Kategori	Skor
Tidak Baik	1
Kurang Baik	2
Baik	3
Sangat Baik	4

Menurut Saifuddin Azwar (2007) , perhitungan skor rata-rata dilakukan dengan rumus 2.4

$$\bar{x} = \frac{\sum x}{n}$$

Keterangan:

\bar{x} = rata-rata perolehan skor

$\sum x$ = jumlah skor yang diperoleh

n = banyaknya butir pertanyaan

Setelah diperoleh rata-rata perolehan skor, kemudian skor tersebut diubah ke dalam kriteria kualitatif dengan mengacu pada tabel 2.6

Tabel 2.6 Rentang Skor (i) kuantitatif

Rentang skor (i) kuantitatif	Kriteria Kualitatif
$\bar{x} > (M_i + 1,50 SB_i)$	Sangat Baik
$(M_i + 0,5 SB_i) < \bar{x} \leq (M_i + 1,50 SB_i)$	Baik
$(M_i - 0,5 SB_i) < \bar{x} \leq (M_i + 0,5 SB_i)$	Cukup Baik
$(M_i - 1,50 SB_i) < \bar{x} \leq (M_i - 0,5 SB_i)$	Sangat Kurang
$\bar{x} \leq (M_i - 1,50 SB_i)$	Sangat Kurang Baik

Saifuddin Azwar (2007)

Keterangan:

X= rata-rata skor tiap butir

M_i = rata-rata ideal = $\frac{1}{2}$ (skor maksimal ideal + skor minimal ideal)

SB_i = simpangan baku ideal = $\frac{1}{6}$ (skor maksimal ideal - skor minimal ideal)

Skor maksimal ideal = skor tertinggi

Skor minimal ideal = skor terendah

Setelah didapat perolehan skor dan kemudian diubah ke dalam kriteria kualitatif maka bisa dilihat apakah sistem tersebut termasuk dalam kategori sangat baik. Baik, cukup baik, sangat kurang atau sangat kurang baik.

G. Web Application Performance Testing (WAPT)

Suatu sistem yang akan diluncurkan perlu dilakukan pengujian terlebih

dahulu agar nantinya dapat dilakukan perbaikan sehingga sistem menjadi lebih baik. Salah satu software yang dapat melakukan pengujian sistem adalah *Web Application Performance Testing* (WAPT).

WAPT adalah *software testing* yang digunakan untuk menguji tingkat stress suatu web. Stress suatu web adalah kondisi dimana web sudah tidak mampu lagi untuk menangani kondisi yang tidak normal (mencakup kuantitas, frekuensi, maupun volume). Ketika akan meluncurkan sebuah sistem berbasis web di internet, maka yang harus dipastikan adalah sistem tersebut mampu menangani pengakses sejumlah yang sudah ditargetkan. Dalam menentukan jumlah pengakses bukanlah hal yang mudah, namun dengan melakukan stress testing hal ini dapat dilakukan. Dengan hasil stress testing tersebut, bisa diketahui performa sistem web dan dapat diperkirakan apakah dengan infrastruktur yang dimiliki layanan akan berfungsi dengan baik atau tidak saat sistem sistem diluncurkan untuk diakses user.

Stress testing terdiri dari 3 yaitu:

1. Performance

Pengujian *performance* meliputi lamanya waktu yang dibutuhkan untuk melakukan semua proses (*processing time*) dan waktu respon server (*response time*) yaitu lamanya waktu yang dibutuhkan pemakai untuk menunggu respon dari server. Menurut penelitian yang dilakukan developer WAPT, ada 3 batas penting untuk nilai waktu respon yaitu:

- 0,1 detik adalah waktu respon yang ideal. Pemakai merasa bahwa sistem bereaksi dengan lancar, dan pemakai tidak merasakan gangguan apapun.

- 1,0 detik adalah waktu respon tertinggi yang dapat diterima. pemakai masih tidak merasa terganggu, meskipun sistem mengalami kelambatan untuk merespon. Namun waktu respon yang lebih dari 1 detik sudah dikategorikan meresahkan pemakai.
- 10 detik adalah batas waktu respon menjadi tidak dapat diterima. Selain itu, berdasarkan penelitian yang dilakukan oleh developer WAPT jika waktu respon melebihi 8 detik, pemakai akan merasa sangat terganggu dan sebagian besar pemakai lebih memilih untuk meninggalkan situs atau sistem tersebut. Biasanya waktu respon harus secepat mungkin. Interval waktu respon yang paling nyaman adalah 0,1 – 1 detik. Meskipun orang dapat beradaptasi dengan waktu respon yang agak lambat. Namun mereka umumnya tidak puas dengan waktu yang lebih dari 2 detik.

2. *Bandwidth*

Pengujian *Bandwidth* meliputi *sent kbit/s* dan *receive kbit/s*. *Sent kbit/s* artinya berapa menghitung banyak kbit per detik yang dapat dikirimkan ke server. Sedangkan *receive kbit/s* artinya menghitung berapa banyak kbits per detik yang diterima oleh server.

3. *Error*

Pengujian *error* meliputi *timeouts on all hits*, *http error on all hits* dan *network error on all hits*. *Timeouts on all hits* menunjukkan persentase tanggapan dengan kesalahan karena timeout dari seluruh halaman. *Http error on all hits* menunjukkan persentase kesalahan karena http dari seluruh halaman. *Network*

error on all hits menunjukkan persentase tanggapan dengan kesalahan karena jaringan dari seluruh halaman.

H. Data dan Alat Penelitian

Berikut ini adalah data dan alat penelitian yang digunakan dalam membuat aplikasi sistem pakar diagnosa gangguan pernafasan:

1. Data Penelitian yang dibutuhkan:
 - Data gejala-gejala penyakit gangguan pernafasan
 - Data nilai MD dan MB yang diberikan oleh pakar
 - Data penyakit gangguan penafasan
 - Data solusi pencegahan pada penyakit gangguan penafasan
3. Alat penelitian dalam pembuatan sistem dibedakan menjadi 2 yaitu perangkat keras dan perangkat lunak. Perangkat keras adalah perangkat pada komputer yang berbentuk fisik dan dapat disentuh. Sedangkan perangkat lunak adalah penerjemah perintah-perintah yang dijalankan pemakai komputer untuk diteruskan atau diproses oleh perangkat keras.
 - a. Perangkat keras:
 - Laptop dengan spesifikasi: Windows 7, Prosesor core i3 2,4 GHz, RAM 2,00 GB, Sistem type 32-bit, Hardisk 500GB
 - Printer
 - b. Perangkat lunak:
 - 1) MySQL
 - 2) PHP

- 3) PhpMyAdmin
- 4) Apache
- 5) Adobe Dreamweaver

1) MySQL

MySQL (*My Structure Query Language*) adalah sebuah program untuk membuat basis data dan bersifat *open source*, artinya siapa saja menggunakannya.

Menurut Didik Dwi Prasetyo (2004), MySQL merupakan salah satu basis data server yang berkembang di lingkungan open source dan didistribusikan secara gratis dibawah lisensi GPL.

MySQL merupakan RDBMS (*Relational Database Management Sistem*) server, RDBMS adalah program yang memungkinkan pemakai basis data untuk membuat, mengelola dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada basis data memiliki relasi antara satu tabel dengan tabel yang lainnya.

Kelebihan dari MySQL adalah memilki bahasa *Query* standar yang digunakan oleh SQL (*Structure Query Language*). SQL adalah bahasa terstruktur yang telah di standarkan untuk semua program pengolah basis data seperti Oracle, SQL Server, dll.

Sebagai sebuah program pembuat basis data, MySQL tidak dapat berjalan sendiri tanpa bantuan aplikasi lain. MySQL dapat didukung oleh hampir

semua program aplikasi baik yang *open source* seperti PHP maupun yang tidak seperti Visual Basic, Delphi dan lainnya.

2) PHP

PHP (*Hypertext Preeprocesor*) adalah bahasa *server-side* programming yang powerfull untuk membuat halaman web yang dinamis dan interaktif. Sintak PHP mirip dengan bahasa Perl dan C. PHP biasanya sering digunakan bersama web server Apache di beragam sistem operasi. PHP juga mendukung ISAPI dan dapat digunakan bersama dengan Microsoft IIS di Windows (Sunyoto, 2007)

Dengan menggunakan PHP maka pengolahan dan manajemen suatu web menjadi lebih mudah. PHP dirancang untuk web dinamis, artinya PHP dapat membentuk tampilan berdasarkan keinginan. Misalnya dapat menampilkan isi database ke halaman web. Penulisan PHP menggunakan script.

Script PHP berkedudukan sebagai tag dalam bahasa HTML (*Hypertext Markup Language*) adalah bahasa standar membuat halaman-halaman web (Kadir, 2001). Untuk menuliskan script php maka harus diapit dengan tag `<?php` sebagai awalnya dan `?>` sebagai akhiran script php. Jika kode yang diketikkan di luar tag tersebut maka tidak akan dianggap sebagai script php oleh PHP *engine*, melainkan akan dianggap sebagai kode html.

Adapun kode berikut adalah contoh kode php yang berada di dalam kode HTML:

```
<html>
```

```
<head>
<title> Latihan menulis PHP </title>
</head>
<body>
Belajar PHP
<?php
echo "Ini adalah bahasa PHP";
?>
</body>
</html>
```

c) phpMyAdmin

Menurut Firdaus (2007), phpMyAdmin adalah suatu program open source yang berbasis web yang dibuat menggunakan aplikasi PHP. Program ini digunakan untuk mengakses database MySQL. Program ini mempermudah dan mempersingkat kerja pemakainya.

Menurut Bunafit Nugroho (2009) phpMyAdmin adalah aplikasi berbasis web yang dibuat dari pemrograman PHP dan diramu dengan javascript. PhpMyAdmin juga dapat disebut sebagai *tools* yang berguna untuk mengakses database mySQL Server dalam bentuk tampilan web. Dengan adanya phpMyAdmin semua pekerjaan yang berhubungan dengan manajemen basis data dan data dalam SQL lebih terbantu.

d) Apache

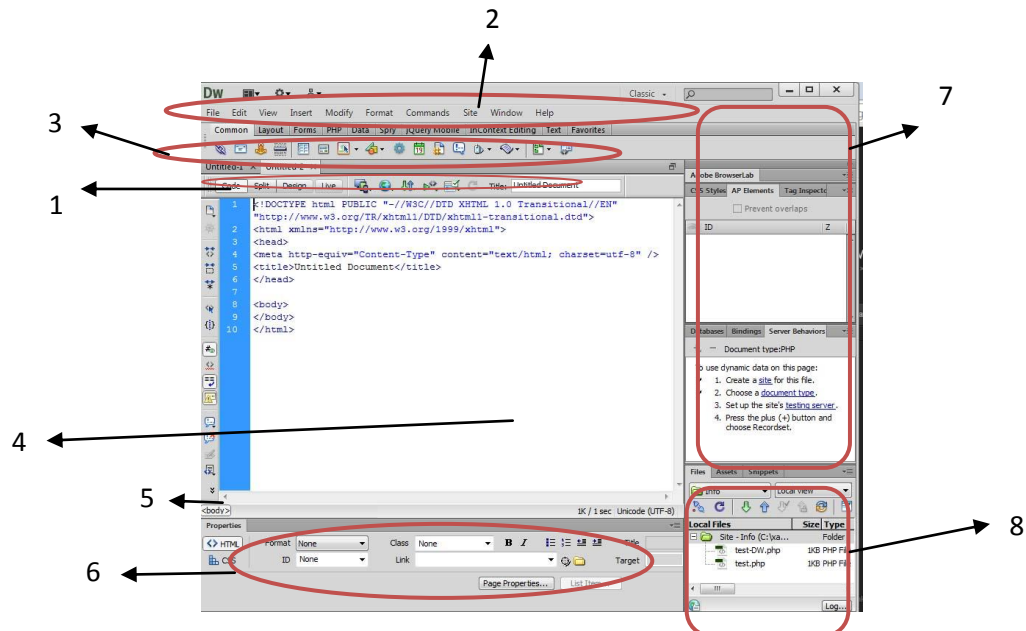
Server HTTP Apache atau server Web/WWW Apache adalah komponen server web dari perangkat lunak LAMP (Linux, MySQL, PHP/Perl/bahasa pemrograman Python) yang dapat dijalankan di banyak sistem informasi (Unix, BSD, Linux, Windows, serta platform lainnya) yang berguna melayani dan mengfungsikan situs web.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pemakai berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah.

e) Dreamweaver

Dalam pembuatan situs web, hal yang tak luput untuk diperhatikan adalah desain web. Perancangan desain web yang menarik dapat membuat pengunjung untuk singgah di web yang dibuat. Salah satu software yang dapat dimanfaatkan untuk merancang desain web secara visual atau software untuk editor web adalah Dreamweaver. *Software* ini cukup populer dikalangan para desainer web, karena kelengkapan fungsinya dan mudah dijalankan.

Gambar 2.12 adalah area kerja dreamweaver (WahanaKomputer, 2009):



Gambar 2.12 Area kerja dreamweaver

1) Toolbar dokumen

Digunakan untuk menampilkan atau mengorganisasikan semua objek yang ada di area kerja. Selain itu juga dapat mengatur tampilan area kerja.

2) Menu Bar

Menu Bar merupakan baris perintah yang terdiri dari menu *popup* yang dapat yang terdiri dari perintah-perintah yang ada di dreamweaver.

3) *Insert Bar*

Berisi barisan ikon-ikon (*shortcutkey*) untuk mempermudah dan mempercepat dalam melakukan penambahan objek di area kerja.

4) *Workspace* / Area kerja

Digunakan untuk menuliskan script kode untuk membuat desain web, selain itu juga menampilkan *live design*.

5) Tag seleksi

Bagian ini terdapat di bawah area kerja. Apabila area kerja sedang kosong maka tag seleksi tidak berisi nilai apa-apa kecuali <body>

6) Properti

Digunakan untuk melakukan perubahan dari objek yang ditampilkan.

Selain itu, juga mengatur properti dari semua objek baik itu berupa image, tabel, animasi atau yang lainnya.

7) Panel Grup

Terdiri dari beberapa panel yang dapat digunakan untuk membuat desain web site.

8) Panel Site

Merupakan salah satu bagian dari panel grub dan berfungsi mengorganisir file-file yang digunakan.

I. Penelitian yang Relevan

Berikut ini beberapa penelitian yang relevan dengan penelitian ini:

- 1) Hamdani (2010). "*Sistem Pakar untuk diagnosa penyakit mata pada Manusia*". Metode yang digunakan dalam penelitian ini adalah *Certainty factor*. Data penyakit yang dikenali menyesuaikan rule yang dibuat untuk dapat mencocokkan gejala-gejala penyakit mata dan memberi nilai persentase agar mengetahui nilai pendekatan jenis penyakit pasien. Untuk mendiagnosa penyakit mata pada manusia tersebut digunakan metode inferensi *foward chaining* dan menggunakan perangkat lunak berbasis *dektop management sistem*.
- 2) Arini Marlyaninrum (2013). "*Aplikasi Sistem Pakar Diagnosis Pada Sistem Komputer*". Pada penelitian ini sistem yang akan dibuat adalah perancangan sistem pakar diagnosis masalah pada komputer dengan menggunakan visual basic. Penulis menggunakan metode prototype serta untuk pemodelan menggunakan UML sedangkan untuk metode inferensinya menggunakan *foward chaining*. Untuk penyimpanan databasenya digunakan ms. access.
- 3) Safian Dhany (2009). "*Perancangan Sistem Pakar untuk Diagnosa Penyakit Anak*". Metode inferensi yang digunakan adalah *Foward chaining*. Dalam sistem pakar ini, ditampilkan pilihan gejala yang dapat dipilih oleh pemakai, dimana setiap gejala akan membawa pemakai ke pilihan gejala berikutnya sampai

mendapatkan hasil akhir. Pada hasil akhir, sistem pakar akan menampilkan pilihan gejala pemakai jenis penyakit yang di derita, penyebab dan penanggulangannya. Di samping itu, sistem pakar ini juga memberikan informasi anak seperti keamanan dan gizi anak. Untuk *user interfacenya* dan pengolahan sistem digunakan visual basic.

4) Andri Pranolo, dkk.(2013). “*Desain Pengembangan Sistem pakar untuk Identifikasi Gangguan Tanaman Hutan dengan Foward Chaining dan Certainty Factor*”. Pada penelitian ini, mengembangkan sistem identifikasi gangguan untuk tanaman kehutanan dengan metode pelacakan *foward chaining* yakni sistem menganalisis persoalan gangguan pada tanaman hutan melalui pencarian fakta yang diinputkan oleh pemakai yang sesuai dalam bagian IF dengan hasil pada bagian THEN dan untuk penarikan kesimpulannya menggunakan faktor kepercayaan (*certainty factor*) atau CF yakni mengkombinasikan nilai CF dari pakar dan nilai CF dari pemakai. Untuk *user interfacenya* dan pengolahan sistem digunakan visual basic.

Penelitian yang dilakukan Hamdani, Arini, Safian dan Andri Pranolo dkk sama-sama menggunakan metode pelacakan *foward chaining*. Menurut, DUR (1994) metode *foward chaining* cocok untuk sistem pakar karena metode ini akan bekerja dengan baik ketika informasi dikumpulkan kemudian dicari kesimpulannya dari informasi yang didapat selain itu metode ini juga menyediakan banyak sekali informasi dari hanya jumlah kecil data.

Untuk metode penarikan kesimpulan Hamdani dan Andri Pranolo menerapkan metode *certainty factor* yang mana pada hasil akhir nya akan ditampilkan

persentase nilai kepercayaan terhadap jenis penyakit yang diderita sesuai dengan gejala yang diinputkan. Namun untuk *user interfacenya* atau tampilannya hamdani memilih untuk menggunakan *dekstop management sistem* sedangkan Andri Pranolo memilih menggunakan visual basic.

Pada penelitian ini, kasus yang akan dibahas adalah diagnosa gangguan pernafasan dengan menerapkan metode inferensi *foward chaining* dan untuk metode penarikan kesimpulannya digunakan *certainty factor* karena metode ini dapat menampilkan nilai persentase kepercayaan terhadap fakta yang diidentifikasi sehingga pemakai melihat seberapa besar nilai keakuratannya. Selain itu, akan diberikan solusi cara penanganan lanjutan sehingga dampak yang buruk diminimalisir. Tampilan sistem berbasis web karena web tidak terbatas oleh waktu dan tempat.